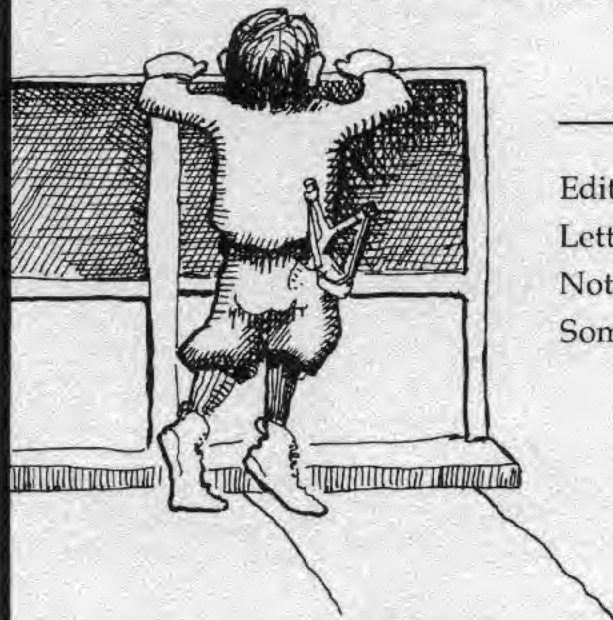


\$3.00



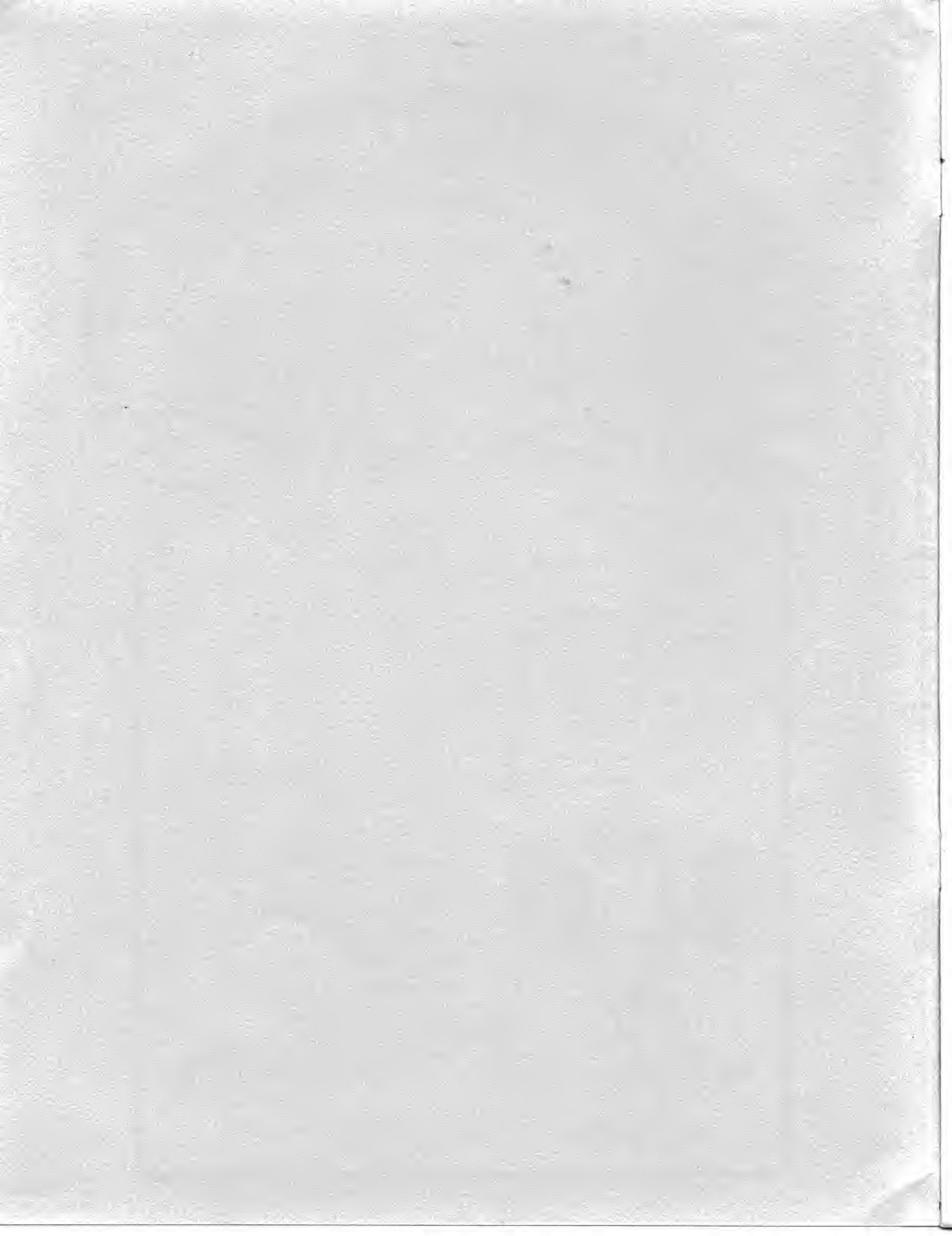
TABLE OF CONTENTS

Supporting a Language	2
Parallel Print Driver & Listing	3
Disk Drive Motor Control	5
Jumpering the Wild Shugart	6
More Power Supplies	7
Direct Input Routine & Listing	7
Program Storage Above PFM & Listing	8



REGULAR FEATURES

Editorial	1
Letters	2
Notes from Garland	4
Something New	7



MICRO CORNUCOPIA
11740 N.W. West Road
Portland, Oregon 97229
503-645-3253

Editor & Publisher
David J. Thompson

Technical Editor
Ruth Fredine-Burt

Graphic Design
Sandra Thompson

Typography
Patti Morris & Martin White
Irish Setter

Cover Illustration
Gerald Torrey

MICRO CORNUCOPIA is published six times a year by Micro Cornucopia of Oregon, 11740 N.W. West Road, Portland, Oregon 97229.

SUBSCRIPTION RATES:

1 yr. (6 issues)	\$12.00
1 yr. (Canada)	\$15.00
1 yr. (other foreign)	\$20.00

All subscription orders payable in United States funds only, please.

ADVERTISING RATES: Available on request.

CHANGE OF ADDRESS: Please send old label and new address.

SOFTWARE, HARDWARE, AND BOOK VENDORS: Micro Cornucopia is establishing a group of reviewers. We would very much like to review your Big Board compatible products for Micro C. Please send material to Review Editor, Micro Cornucopia.

WRITER'S GUIDELINES: All items should be typed, double-spaced on white paper or better yet, on disk. (Your disk will be returned promptly.) Payment is in contributor's copies.

LETTERS TO THE EDITOR: Please sound off.

CP/M is a trademark of Digital Research, Inc.

Copyright 1981 by Micro Cornucopia.
All rights reserved.

MICRO CORNUCOPIA

Sept. 1981

The Journal of the Big Board Users

No.2



*There once was a
Big Board so brisk.
It could eat all the
bits off a disk.
It chewed up the bits,
then spit out the pits,
which made feeding it
software a risk.*

Here We Go Again!

Exclusive!

What happens when a Xerox copies a Big Board? Why you get a "Worm", of course! That's right! The Xerox 820 is just a Big Board in disguise.

My informed sources say that last fall Xerox bought non-exclusive rights to manufacture a system based on the Big Board. Xerox re-laid out the board (4 layers) so that it would fit in the cabinet, they dedicated the SIO port B as a printer port, and they set up the disk interface (1771) to handle either 5 or 8 inch. Otherwise, it appears to be all Big Board, right down to the 2.5 MHz clock. The system PIO does the same things on both systems, bit for bit, according to Xerox's documentation.

Xerox had 50,000 orders in hand the day they shipped the first 820, and they expect to recoup all their startup costs by the end of this calendar year. What a market for software and hardware developed around the Big Board. I'll say more about the 820 as information comes in. (I'd give my eye teeth to see a schematic and service manual for the 820.)

Picnic

We had a Saturday noon picnic to celebrate our first issue. It turned out that the Saturday we picked conflicted with every party/birthday/outing/etc. for three states around. But Sandy and I and those who came had six hours of very interesting and mellow conversation.

The knowledge, resources, and excitement among the local group members are terrific. I only wish all of you could have joined us.

The First Issue

Despite the speed of the U.S. Snail, a heartening number of readers have actually received issue no. 1. The responses from these lucky folks have made the daily trip out to our mailbox most enjoyable. The comments have included; 'surprised, happy, delighted'.

Though Micro C is a long way from being a success financially, feedback like this tells us that it is successful in other ways. We like doing it and we really appreciate your response.

Sometimes a dream generates momentum of its own. This one has.

Thanks.

David Thompson
Editor & Publisher

Letters

Dear Sir,

July came and July went by, and my mailbox has completely rusted out due to all that drooling.

Silly me! When I read 'Issue No. 1 will hit the streets during July' I assumed it was July 1981! But now I realize you meant July 1982. I'd better get a stainless steel mailbox or maybe not bother to wait, because the magazine will never get here.

Maybe it went the way of Mitt's Newsletter, the Digital Group Newsletter, and Processor Technology's "Access."

I hope not.

Joe Kish
758 Yucca Ridge Lane
San Marcos, CA 92069

Editor's note:

I called Joe; after all it was the least I could do for his mailbox. And besides, I think it's a great letter! (He did finally receive issue no. 1.)

Sandy and I made a desperate, last ditch effort to get all 500 first issues collated, bound, labeled, sorted and bundled in one afternoon so we could get the first issue in the mail on July 31. We missed the 8 PM deadline at the post office by 15 minutes.

So the magazine was mailed Monday morning, August 3rd. (So much for hitting the streets in July.)

Someday maybe I'll write a book about starting a users group magazine. I could almost write the book about the first issue, and Murphy would certainly be a leading figure. (For those of you who don't know Murphy, he is the one credited with the first voyage of the Titanic.)

Quote from Murphy:

**If there is no way
your plan can fail,
you simply don't have
all the information.**

Dear Editor,

I bought a bare board version and built it up from scratch. I had to buy about \$80.00 worth of parts beyond what I had around. I have it up and running CP/M and am currently working on packaging it in a terminal-type case with a Ball Brothers CRT. The unit is going to be used for text processing and formatting for a friend's photo typesetter. My other computer is an LSI-11 and I also use

(continued next column)

Supporting A Language

By David Thompson

Throughout these early months of Micro Cornucopia, I have been looking at commercial and public versions of various languages with the hope of finding a semiofficial language for this group.

A common high level language would mean we could pass around source code in something other than assembler. But the language would need to be powerful enough for substantial commercial applications and inexpensive enough that most of the people in the group could afford it.

Letters continued

my H19 with the DEC-20 at work. I think the Big Board is an excellent value and very useful.

I agree that Frank Gentges' idea about the parallel ports is excellent. That would take care of most of the board's limitations. I think your publication has already been worth the price and I suspect that an active users group with a publication will enhance the usefulness of the hardware significantly.

Doug Faunt
PO Box 11142A
Palo Alto CA 94306

Dear David,

CONGRATULATIONS!!! FANTASTIC!!! You really made it. It looks great and reads great. You are certainly to be congratulated for undertaking such a task that should be helpful to so many.

I hate to mention that Momma and I are just back from five weeks vacation in the Smokey Mountains in Tennessee. I am about ready to get my feet on the ground again. I hope that I can get back on track to help keep the pipe full of articles for future issues.

Don Retzlaff
6435 Northwood
Dallas TX 75225

Editor's note,

What can I say? Thanks again Don, without you and John Jones and Andrew Beck, and the rest of you who are writing up things for future issues this wouldn't be possible. (As for the five whole weeks in the Smokey Mountains, that's just not fair.)

Plus, it would need to produce fast and compact object code, encourage readable source code, and promote structured programming. (Whew!)

I am looking seriously at three languages: Forth, Pascal, and C. Of these three, C is presently leading. One reason is that all the versions I have seen have been upwardly compatible with Bell Lab's C.

Versions of C that I'm aware of:

Small C (Public)
Small C+ (Public)
Tiny C (\$100)
CW/C (\$75)
BDSC (\$145)
Supersoft C (\$200)
Whitesmith's C (\$600)

(The prices are approximate.)

Whitesmith's C is a full blown version of the language. In fact, sources tell me that it was created by three fellows who worked on C for Bell Labs. They left Bell in order to develop and market C for the business and scientific community.

I've heard that BDSC is a competent enough subset to be an option for someone writing commercial applications. It has its own users group and publication. All this for \$145, such a deal. (Lifeboat is offering discounts on quantity purchases of BDSC.)

CW/C is an expanded version of Small C with lots of nice utilities, but I don't know if it is ready to do commercial work. However, it still looks like quite a bargain at \$75.

Tiny C is the only interpreter in the bunch. It also comes in compiler form for about \$300. The only thing I have heard about Tiny C is that it has an excellent manual (and I heard that fourth or fifth hand).

Supersoft's C is new on the market. The ads say that they support 'most' of version 7 Unix. If that includes floating point and pointer arithmetic, then it would be a very credible piece of software, assuming they have taken time to exorcise bugs.

The standard text on C is:

"The C Programming Language"
by Kernighan and Ritchie
Prentice-Hall

Notes From Garland, Texas

By David Thompson

Clearing up the screen.

The clear-to-end-of-screen command is CONTROL Q, not CONTROL W as indicated in the documentation.

Bringing up stubborn boards.

A number of people have been contacting Jim and me about problems they are having bringing up boards. One of the most common symptoms is a pattern of two characters on the screen or a screenful of random garbage. Either way, it basically means that the board probably didn't finish loading the PFM monitor in RAM so it could try to clear the screen.

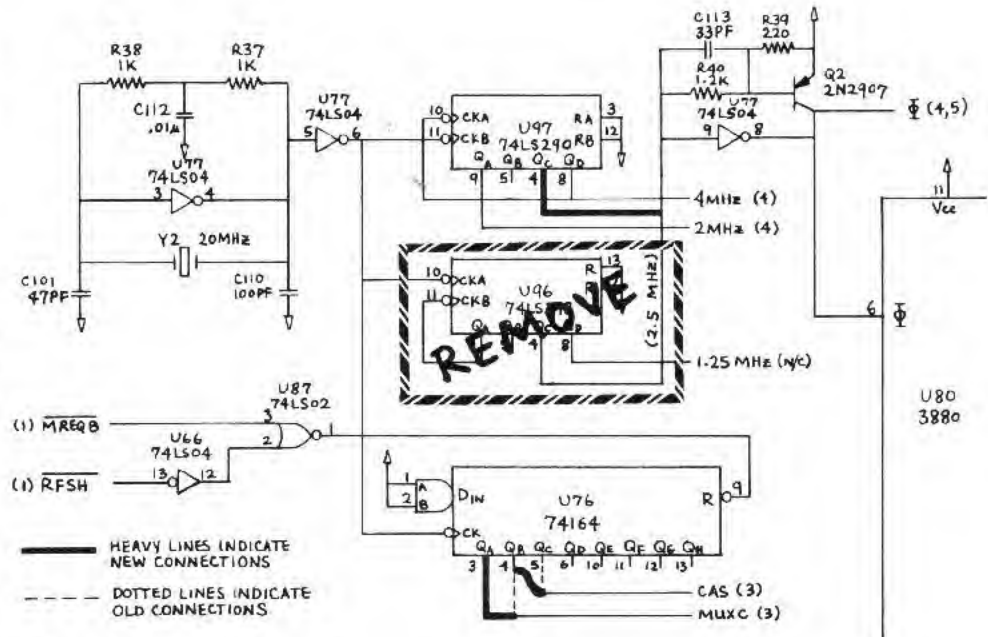
Jim is going to put together information about what they look for when they troubleshoot boards. Hopefully, I will have that in time for the next issue.

Don't forget the 90 day guarantee which completely covers defective parts and boards. Plus, he has been doing out-of-warranty or pilot error repairs very reasonably. Most of the time these charges have been between \$25 and \$50. The maximum so far has been \$75 (the board had to be almost completely resoldered, among other things). That's pretty hard to beat.

Two CP/Ms

I have noticed that some software which runs on one Big Board system will not necessarily run on another. I also noticed that there are two different IDs when CP/M boots.

I called Jim about this and he said that those folks who used the BIOS he sent out with the boards and who did their own incorporation into CP/M have a version which originates the BIOS at EA00. All the folks who bought CP/M already modified for the Big Board have a BIOS starting at E800. The difference has led to some problems with software which depends on having BIOS in a certain place.



4 MHz Modification Version 2

Jim said the ready-to-run version has BIOS shifted down 200H because they thought they needed room to store 256 bytes (a double-density sector) in high memory. Then the data could be moved into low memory in 128 byte chunks and accessed. Jim isn't sure whether there is going to be a use for this space but he is concerned that we maintain consistency.

According to Jim, it's easy to make the EA00 BIOS into an E800 BIOS.

Original— .RES.(MSIZE-20)*1024

New— .RES.((MSIZE-20)*1024)-200

Now reassemble the mess and you too can ORG at E800.

By the way, a pretty reliable way to tell which version you have is to look at the ID that's displayed when you boot CP/M. If it just says "60k CP/M version 2.2" then you probably ORG at EA00. If the prompt includes the words "BIG BOARD" then you already ORG at E800.

The separate BIOS (and monitor etc.) disk Jim is shipping with orders now ORGs at E800. If you would like the latest version rather than reassembling BIOS with the modification above, send Jim a disk and \$3.00 for shipping.

4 MHz (Again).

This is an updated version of the 4 MHz mod printed in issue no. 1. This version reportedly does not require special ram. Jim says he has 300ns 4116 working consistently using this mod. The only difference between this one and the previous one is that the CAS and MUXC lines are each moved left one pin on U76 (shift register) so that they change states 50ns earlier. This change means that the system meets the precharge requirements for the slower RAM.

4 MHz Mod Version 2

1. Cut the trace (bottom of the board) to U76 pin 4.
2. Connect the cut trace (MUXC) to U76 pin 3.
3. Cut the trace (bottom of the board) to U76 pin 5.
4. Connect the cut trace (CAS) to U76 pin 4.
5. Remove U96.
6. Connect U97 pin 4 to U96 pin 4.
7. Don't replace U96.

(continued next page)

Disk Drive Motor Control

By David Thompson

CP/M patch for serial printer port.

This CP/M modification redirects the list device output to serial port B. The default data rate is 300 baud. This patch does not force the Big Board to poll any of the handshake lines on port B. Thus, it has no way of knowing if the printer buffer is full. (May or may not be a problem.) This modification is for those who ORG at E800.

Enter the characters inside the quotation marks. <CR> = carriage return.

The patch:

1. Power up the Big Board (BB).
2. Place a CP/M disk with SYSGEN on it, in drive A.
3. Boot CP/M.
4. Enter "SYSGEN" "<CR>"
Displays: SYSGEN VER. 2.0
Displays: SOURCE DRIVE NAME...
5. Enter "A"
Displays: SOURCE ON A,
THEN TYPE RETURN
6. Enter "<CR>"
Displays: FUNCTION
COMPLETE...
7. Hit the BB RESET switch <CR>

NOTE: You now have an image of Boot, CP/M, and Bios in RAM starting at 0900H.

8. Remove the source disk from drive A.
9. Enter "M22C7" "<CR>"
Displays: 22C7 00
10. Enter "79"
11. Enter "C3"
12. Enter "18"
13. Enter "F0"
14. Hit spacebar to return to PFM.
15. Enter "M1F90" "<CR>"
16. Enter "47"
17. Enter "EB"
18. Hit spacebar to return to PFM.
19. Place blank disk in drive A.
20. Enter "G100"
Displays: SYSGEN VER 2.0
21. Enter "<CR>"
Displays: DESTINATION
DRIVE...
22. Enter "A"
Displays: DESTINATION
ON A...

23. Enter "<CR>"
Displays: FUNCTION
COMPLETE...
24. Enter "<CR>"

The disk now contains a CP/M system that supports CONTROL P (and PIP LST:=) for listings. As mentioned above, the output is on serial port B and is 300 baud.

Editor's note:

To change the baud rate, create F.COM as follows:

1. Enter "DDT" "<CR>"
2. Enter "A100" "<CR>"
3. Enter "MVI A,XX" "<CR>"
4. Enter "OUT 0C" "<CR>"
5. Enter "JMP 0" "<CR>"
6. Enter "<CR>"
7. Enter "G00" "<CR>"
8. Enter "SAVE 1 F.COM" "<CR>"

This routine sends a single byte (XX) to the channel B baud rate generator. I am working at 9600 baud so I replace XX with 0E. See the Big Board Theory of Operation for other baud rates.

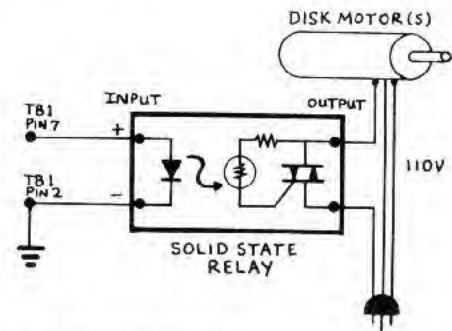
Once you have completed the baud rate program, simply enter "F" "<CR>" from the CP/M prompt to set the baud rate.

No UPS to a PO Box?

Jim Tanner lists his mailing address as a PO Box but he also has a street address that works for both the post office and United Parcel Service. (The ZIP is different.)

Jim Tanner
Digital Research Computers
2702 Industrial Lane
Suite J2
Garland, Texas 75041
Phone 214-271-3538

■ ■ ■



Disk AC Control Circuit.

If you're tired of listening to your disk drives grind on hour after hour, here's relief.

The board must have the timer option installed and you must jumper pin 3 to pin 4 and pin 7 to pin 8 on JB2. This supplies the one second interrupt to the Z80. If the Z80 counts all the way to 30 after the most recent disk access then it sends a command to the system PIO to drive the output of U112 pin 2 low.

Terminal 7 on the Big Board power connector is tied to U112 pin 2. This terminal is high (about 4V) when the system is doing a disk access and goes low if there hasn't been an access for 30 seconds.

Simply connect the input of an optically isolated solid state relay between terminal 7 and ground. Then connect the output in series with the AC to the disk drive motors. (But do not connect in series with the drives' DC supply.)

I tried mechanical relays at first, but even the type made to be driven by TTL have problems. Whenever you use mechanical switches to start and stop motors you get interesting transients on the AC line. Interesting transients occasionally cause CPUs to go off picking daisies.

I am now using an ITT solid state relay P6-3DCC-120R5. It has a (P6) package, a 3VDC (3D) input, a 120VAC output with random switching point (120R), and it handles up to (5) amps. It is also small, quiet, and hasn't yet sent the system packing.

■ ■ ■

Jumpering The Wild Shugart

By David Thompson

Shugart set a new standard for obscurity when they came out with their SA 801 user's manual.

It's not that they don't tell you how to jumper their drives, the only problem is figuring out what they told you. Once you figure it out, don't go back and look at the manual, you'll just get confused again.

So on that note, here's what I figured out.

For drive A, jumper only the following: DC, C, DS1 (Drive Select 1), T2, T3, T4, T5, T6, HL, A, B, T1, 800, Y.

For drive B, change DS1 to DS2. For drive C, change DS1 to DS3, and so on.

For the last 9 months or so, Shugart has been shipping drives with a new circuit board. The new board is completely interchangeable with the old one, but the new one does not use the -5/-15V pin on the DC supply jack (J5). The pin is there but is not connected to anything because the new board does not need -5V.

One way to tell whether you have a new or old style drive is to check the bottom left hand corner on the circuit board. The old drive has a -5V regulator there. On the new one, that corner is pretty empty. Also, the resistance from the -5V pin to ground is infinite on the new boards.

I had one of the new boards but the old documentation so I spent a couple of 'interesting' evenings trying to make sure the -12V I was supplying would be properly turned in-

to -5V on the board. (Oh well, if everyone's documentation were perfect there probably wouldn't be so much need for user groups.)

Note: The following information is from Bill Klevesahl, Shugart's product manager for the SA 800 series.

Test points for both boards.

- 1,2 Amplified read signal
- 5,6,7 Ground
- 10 -Index
- 11 +Head Load
- 12 -Index and Sector Pulses
- 16 +Read Data
- 25 +Write Protect
- 26 -Detect Track 0
- 27 +Step Pulse

Test points on the old board only.

- 3,4 Differential Read Signal (this signal is now hidden inside the new LSI read chip).
- 21,24 -Data Separator Timing (there is no longer a pot to adjust this).

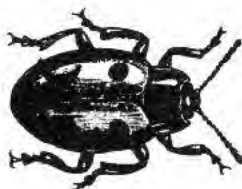
Test points on the new board only.

- 8 +Data Window (for checking FM data separation).

Optional features on the new board.

- Add-trace option TS enables true FM data separation, maintaining synchronization during address marks.
- Add-trace option NFO prevents the head from being forced out past track 0.

BUG



The formatting program listed in issue 1 contains a bug. If the program has a problem accessing a disk in drive B, it reformats the disk in the default drive (A).

Issue 3 will include a revised format program.

Coming Up

Articles you'll be seeing in the future.

- Reverse video cursor
- 5 inch disk interface
- Real time clock routine
- Converting a TV into a real video monitor
- More on the PFM monitor
- Review of 3 assembly language texts
- Bios modifications

Articles we'd love to see.

- Trials and tribulations of bringing up a Big Board
- How you've improved the PFM monitor
- Hard disk interface
- Filling out the second bank with system RAM
- DMA interface
- Double density disk interface
- A graphics display
- A speech generator
- A simple ROM burner
- Interfacing with particular printers etc.
- An in-depth series on CP/M
- Reviews of FIG Forth and Forth 79
- Reviews of BDSC, Whitesmith's C, CW/C and Super-soft's C
- Computer consulting using a Big Board
- Reviews on peripherals, keyboard, video monitor, power supply, cabinet, disks, etc.
- Other software reviews. Even if you are just borrowing a copy to evaluate, please let us know how you like it.
- Book reviews

If you are immersed in any of these projects, please share your experience with all of us.



Direct Input Routine

By Andrew P. Beck

AB Computer Products
PO Box 571
Jackson, NJ 08527

Assembly Listing

```

F800  E5      SUBR    PUSH HL      ;SAVE ADDRESS OF HL%
F801  CD06F0  CALL KBDST ;GET KBD STATUS
F804  B7      OR A      ;IF A=0 DATA AVAILABLE
F805  CA0EF8  JP Z ISDATA ;JP TO DATA SAVE ROUTINE
F808  E1      POP HL     ;GET ADDRESS BACK
F809  3C      INC A      ;A=FF IS NO DATA, MAKE IT 0
F80A  77      LD (HL),A   ;STORE 0 IN HL%
F80B  23      INC HL     ;DO BOTH BYTES
F80C  77      LD (HL),A
F80D  C9      RET        ;RETURN WITH HL% = 0
F80E  CD09F0  ISDATA CALL KBDIN ;GET INPUT CHAR INTO A
F811  E1      POP HL     ;GET ADDRESS OF HL% BACK
F812  77      LD (HL),A   ;STORE DATA, LOW ORDER
F813  23      INC HL
F814  3600    LD (HL),0    ;HIGH ORDER = 0
F816  C9      RET        ;RETURN TO BASIC
    
```

-- Poke the above program into F800+ --

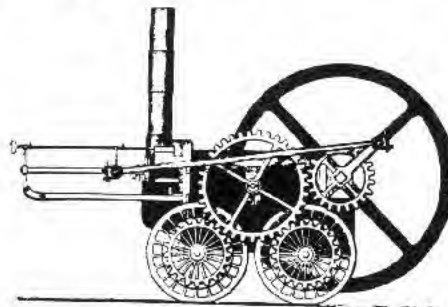
```

500 SUBR = %HF800
510 DATA %HES,%HCD,%H06,%HFO,%HB7,%HCA,%HOE,%HFB
520 DATA %HE1,%H3C,%H77,%H23,%H77,%HC9,%HCD,%HO9,%HFO
530 DATA %HE1,%H77,%H23,%H36,%H00,%HC9
540 FOR I=0 TO 22
550 READ INST
560 POKE SUBR+I, INST
570 NEXT
    
```

-- Demonstration routine --

```

580 HL%=0
590 CALL SUBR (HL%)
600 IF HL%=0 GOTO 590
610 IF HL%=3 THEN STOP
620 PRINT CHR$(HL%);
630 GOTO 590
    
```



This routine makes it possible to do direct input with Microsoft basic. First, a machine language subroutine is poked into an unused area of the system monitor.

This subroutine calls the monitor subroutine and the monitor checks to see if an input character is available. If none is available, the IIL% is set to zero. If a character is available, it is stored in HL.% before a return is executed.

In the demonstration program, a returned character is echoed on the console. If the character is ^C, the demonstration stops.

■ ■ ■

More Power Supplies

By David Thompson

I just received a catalog from ACDC Electronics and they list a power supply that should power the Big Board and a couple of drives. (Like the Power One, you still have to finagle +12V but that isn't hard, see Issue no. 1.)

Model ETV801 provides:

- +5V at 9 amps
- 12V at 0.8 amps
- +24V at 4.5 amps peak

Price is \$132 (list, single)

They don't mention how they handle over-current protection, but they do indicate that they only have over-voltage protection on the +5V line unless you specify the -1 option. They don't say how much extra you pay for the option.

ACDC Electronics
401 Jones Rd
Oceanside, CA 92054

Power/Mate also has an open frame linear with the same specifications as the ACDC model above, but the PowerMate model ED-132AV lists for \$120 (single).

Power/Mate
514 S River St
Hackensack, NJ 07601

■ ■ ■

Something New

DataCast
345 Swett Road
Woodside, CA 94062

I just received issue no. 1 of DataCast and I'm impressed, very impressed. This is a bimonthly magazine for 'major micro systems and telecommunications.' 'Major micro systems' means CP/M in a business or OEM environment and 'telecommunications' means networking.

Jim Warren, guiding force behind the West Coast Computer Faire, is behind this magazine and I suspect it will be around for a long while. Subscriptions are \$18 per year (6 issues).

He is starting with a staff of 19 (if you include the mascot, Sir Lick-A-Lot) and it shows. The first issue is

64 pages and about 60 pages of that is copy.

Some first issue articles:

- What is Telidon and Why is AT&T Adopting It?
- Overview of Home Information Services
- A Seminar for Independent CP/M Software Vendors
- Software Documentation Protocols
- An Index to CP/M Software and Vendors

Other Interesting Periodicals

Dr. Dobb's Journal
PO Box E
Menlo Park, CA 94025

Lifelines
1651 Third Ave
New York, NY 10028

Please let us know about your favorite magazines.

Program Storage Above PFM

By Don Retzlaff

6435 Northwood
Dallas, TX 75225

There are numerous times when you want to write a small assembly language program to use as a printer driver or other routine. These small utilities need to reside in high memory so they can operate at the same time as routines which reside in the normal transient program area (starting at 0100H).

Since programs are loaded starting at 0100H, these utilities must load themselves into high memory.

There is a considerable amount of memory available above PFM that is not dedicated to any other use. PFM version 3.3 uses upper memory starting at F000H through F7E6H. The RAM area FF00H through FFC8H is used for data storage. This leaves the memory from F7E7H through FEFFH and FFC9H through FFFFH available for your use. Not all of this space is really available since future releases of PFM could use some of this space.

I recommend that you limit your programs to the following areas: (FA00H through FEFFH and FFE0H through FFFFH).

Moving the program up

In order for your routine to start out as a normal COM file but wind up in upper memory, it has to do a quick shuffle.

1. When the COM file is executed it is loaded into memory starting at 0100H.
2. Execution starts at 0100H.
3. The first few statements (starting at 0100H) must copy the routine into upper memory.
4. An initialization routine may then be executed.
5. Control is then transferred to the routine or back to PFM.

In order to accomplish all of the above it is necessary to do the following:

1. Write your assembly language routine as follows:
 - a. The origin is set at the desired point where your routine is to reside.
 - b. Your program must start with a short move routine.

- c. An initialize routine usually follows that patches (hooks) your routine into the monitor or PFM.
- d. Your routine follows.
- e. The last statement defines the length of the program.

2. Assemble your program.
3. Execute DDT and load your HEX file into memory. Typically this is done as follows:

>A DDT NAME.HEX

This will load your program into memory at the desired location (example EA00H). The program will not execute.

DDT will print out starting and ending addresses.

NEXT PC/n
FAxx FA00

4. Using DDT, move the program from upper memory to 0100H.
MFA00,FAxx,0100
5. Transfer control back to PFM by typing:
G0
6. Save the program using the SAVE command.

SAVE 1 NAME.COM

You must save the program in 256 byte blocks. Using '1' will save 256 bytes, '2' would save 512 bytes, etc.

7. The program is now ready for execution as a COM file.

The above procedure may seem long and rather involved but after you have done it a few times you will find it very quick and simple.



PFM Monitor Listing (continued from issue no. 1)

F37B	C0	0745	NZ	ERROR IF > 4 NUMBERS ENTERED	F3E3	C690	0822	ADD	A, 90H
F37C	C5	0746	BC	:SAVE PARAMETER COUNT	F3E5	27	0823	DAA	
F37D	CD9FF3	0747	BETHX	:READ A NUMBER FROM LINE BUFFER	F3E4	CE40	0824	ADC	A, 40H
F380	C1	0748	BC		F3E8	27	0825	DAA	
F381	D8	0749	BC		F3E9	C315F4	0826	JP	OUTPUT
F382	DD217CFF	0750	C				0827		
F386	DD09	0751	IX,PARAM1	:ERROR IF RESULT OVER 16 BITS			0828		
F388	DD7500	0752	IX,BC	:POINT TO PARAM STORAGE AREA			0829		
F38B	DD7401	0753	(IX+0),L	:ADD PARAMETER COUNT IN BC			0830		
F38E	FE20	0754	(IX+1),H	:STORE DATA RET FROM 'GETHEX'			0831		
F390	2BE4	0755	2,PARAM1-\$				0832		
F392	FE2C	0756	2,PARAM1-\$:GET ANOTHER ITEM IF SPACE			0833	EDT	04H
F394	2BE0	0757	2,PARAM1-\$:GET ANOTHER ITEM IF COMMA			0834	CR	0DH
F396	FE0D	0758	CR				0835	LF	0AH
F398	37	0759	SCF				0836		
F399	C0	0760	NZ	:ELSE CHECK FOR CARRIAGE RETURN			0837		
F39A	79	0761	A,C	: AND EXIT WITH CY=1 IF NOT			0838	PNEXT:	(SP),HL
							0839	CALL	PM5G
							0840	EX	(SP),HL

```

F39B DB3F      0762      A      :A=COUNT OF NUMBERS ENTERED
F39D 3C        0763      A
F39E C9        0764      RET
;
0765      :
0766      :GETEX CONVERTS ASCII TO BINARY AND DOES
0767      :HIGH LIMIT CHECKS TO LESS THAN 17 BITS.
0768      :CARRY SET ON ILLEGAL CONVERSION RESULT
0769      :TERMINATING CHARACTER RETURNS IN A.
0770      :HL RETURNS WITH 16 BIT BINARY INTEGER
0771      :
F39F 210000    0772      LD      HL,0
F3A2 180B      0773      JR      GNUM3-$
0774
F3A4 0604      0775      GNUM1: LD      HL,HL
F3A6 29        0776      GNUM2: ADD     C,B,4
F3A7 DB        0777      RET     C
F3A8 10FC      0778      D1NZ   GNUM2-$
F3AA 5F        0779      LD      E,A
F3AB 1600      0780      LD      D,0
F3AD 19        0781      LD      HL,DE
F3AE DB        0782      ADD     C
F3AF FD7E00    0783      GNUM3: LD      A,(1Y+0)
F3B2 FD23      0784      INC     IY
F3B4 4F        0785      LD      C,A
F3B5 CDBDF3    0786      CALL   ASCHEX
F3B8 30EA      0787      JR      NC,GNUM1-$
F3BA 79        0788      LD      A,C
F3BB B7        0789      OR      A
F3BC C9        0790      RET
0791      :
0792      :
0793      :ASCHEX: SUB     '0'
0794      :
0795      :
0796      :
0797      :
0798      :
0799      :
0800      :
0801      :
0802      :
0803      :
0804      :
0805      :
0806      :
0807      :PUT4HS: LD      A,H
0808      :
0809      :
0810      :
0811      :
0812      :
0813      :
0814      :PUT2HX: PUSH   AF
0815      :
0816      :
0817      :
0818      :
0819      :
0820      :
0821      :PUTNIB: AND     00001111B

```

```

F3F1 C9        0841      RET
0842      :
0843      :MSG: LD      A,(HL)
0844      :
0845      :
0846      :
0847      :
0848      :
0849      :
0850      :
0851      :CRLF OUTPUTS A RETURN-LINEFEED-SPACE
0852      :TO THE CONSOLE DEVICE
0853      :
0854      :CRLFS: CALL   PNEXT
0855      :
0856      :SPACE: DEFB   CR,LF,EOT
0857      :
0858      :
0859      :
0860      :
0861      :ECHO INPUTS ONE CHARACTER FROM THE CONSOLE
0862      :DEVICE, PRINTS IT ON THE CONSOLE OUTPUT AND
0863      :THEN RETURNS IT IN REGISTER A WITH BIT 7 RESET
0864      :
0865      :OUTPUT PRINTS THE CHARACTER IN REGISTER A ON
0866      :THE CONSOLE OUTPUT DEVICE AND THEN DOES A CHECK
0867      :FOR CONSOLE INPUT TO FREEZE OR ABORT OUTPUT.
0868      :
0869      :
0870      :
0871      :
0872      :
0873      :
0874      :
0875      :
0876      :
0877      :
0878      :
0879      :
0880      :
0881      :
0882      :
0883      :
0884      :
0885      :
0886      :
0887      :
0888      :
0889      :
0890      :
0891      :
0892      :
0893      :
0894      :
0895      :
0896      :
0897      :

```

INCLUDE INTSRV.ASM

PFM Monitor Listing (continued)

```

0898 :*****
0899 :*
0900 :* INTERRUPT SERVICE ROUTINES FOR KEYBOARD
0901 :* INPUT AND REAL-TIME CLOCK FUNCTIONS
0902 :* 3-Aug-80
0903 :*
0904 :*****
0905 :*
0906 :*
0907 :*
0908 :*
0909 :* KBDST: LD A, (FIFONT) ;GET INPUT FIFO BYTECOUNT
0910 :* OR A ;TEST IF EQUAL ZERO
0911 :* RET Z ;EXIT WITH A=0 IF QUEUE EMPTY
0912 :* LD A,255
0913 :* RET
0914 :*
0915 :*
0916 :*
0917 :* KBDIN: CALL KBDST
0918 :* JR Z,KEDIN-$ ;LOOP UNTIL KEYBOARD INPUT RDY
0919 :* PUSH HL
0920 :* CALL REMOVE ;GET CHARACTER FROM INPUT QUEUE
0921 :* POP HL
0922 :* RET
0923 :*
0924 :*
0925 :*
0926 :*
0927 :*
0928 :* STASH: LD HL,LOCK ;POINT TO SHIFT LOCK VARIABLES
0929 :* CP (HL) ;TEST IF A=SHIFT LOCK CHARACTER
0930 :* INC HL ;THEN POINT TO LOCK FLAG
0931 :* JR NZ,STASH2-$ ;JUMP IF NOT SHIFT CHARACTER
0932 :* INC (HL) ;ELSE COMPLIMENT THE SHIFT LOCK
0933 :* RET ;AND EXIT NOW
0934 :*
0935 :* STASH2: BIT 0,(HL) ;TEST THE SHIFT LOCK FLAG
0936 :* JR Z,STASH3-$ ;JUMP IF SHIFT LOCK NOT SET
0937 :* CP 40H ;ELSE CHECK FOR SHIFTABLE CHAR
0938 :* JR C,STASH3-$ ;AND JUMP IF NOT = OR GREATER
0939 :* CP 7FH ;THAN 'a' AND LESS THAN RUBOUT
0940 :* JR NC,STASH3-$
0941 :* XOR 00100000B ;ELSE TOGGLE BIT 5 OF THE CHAR
0942 :* LD C,A
0943 :* LD HL,FIFONT ;BUMP INPUT FIFO CHAR COUNT
0944 :* LD A,(HL)
0945 :* INC A
0946 :* CP 16
0947 :* RET NC
0948 :* LD (HL),A ;EXIT NOW IF FIFO IS FULL
0949 :* LD HL,FIFIN ;ELSE INCREMENT FIFO COUNT
0950 :* CALL INDEX ;POINT HL TO FIFO INPUT OFFSET
0951 :* LD (HL),C ;STORE CHARACTER IN FIFO @ HL
0952 :* RET
0953 :*
0954 :*
0955 :*
0956 :*

```

```

F4C1 CDE7F4 1021 DSFTH: CALL CALLHL ;CALL SUBROUTINE ADDRESSED BY H
F4C4 F1 1022 POP AF
F4C5 C1 1023 POP BC
F4C6 D1 1024 POP DE
F4C7 E1 1025 POP HL
F4C8 ED7B35FF 1026 LD SP,(SPSAVE) ;RE-ENABLE INTERRUPTS & RETURN
F4CC FB 1027 EI
F4CD ED4D 1028 RETI
1029 :
1030 :
1031 : -- RX ERROR INTERRUPT SERVICE ROUTINE FOR SIO --
1032 :
1033 : ARRIVE HERE IF RECEIVE INTERRUPT FROM FRAMING. OVERRUN
1034 : AND PARITY ERRORS. (PARITY CAN BE DISABLED)
1035 :
1036 : SIOERR: LD (SPSAVE),SP ;SAVE USER STACK POINTER AND
F4D3 3157FF 1037 LD SP,IMPSTK+32 ; SWITCH TO LOCAL STACK
F4D4 F5 1038 PUSH AF
F4D7 DB5F4 1039 CALL SIOIN2 ;CLEAR BAD CHARACTER FROM SIO
F4DA 3E07 1040 LD A,'G'-64
F4DC DD15F5 1041 CALL SIOXMT ;OUTPUT A CTL-G AS A WARNING
F4DF F1 1042 POP AF
F4E0 ED7B35FF 1043 LD SP,(SPSAVE)
F4E4 FB 1044 EI
F4E5 ED4D 1045 RETI
1046 :
1047 :
1048 : CALLHL: JP (HL)
1049 :
1050 :
1051 :
1052 : POLLED MODE I/O ROUTINES FOR SIO CHANNEL B
1053 :
1054 : SIOST: IN A,(SIOCPB) ;GET SIO STATUS REGISTER
1055 : AND 00000001B
1056 : RET Z
1057 : LD A,255 ;ACC=0 IF NO DATA AVAILABLE
1058 : RET
1059 :
1060 :
1061 : SIOIN: CALL SIOST ;TEST CONSOLE STATUS
1062 : LD 2,SIOIN-$ ;LOOP UNTIL DATA IS RECEIVED
1063 : LD A,00110000B ;RESET STATUS BITS IN SIO FO
1064 : OUT (SIOCPB),A ;PARITY/OVERRUN/FRAMING ERRORS.
1065 : IN A,(SIODPB) ;THEN GET THE INPUT CHARACTER
1066 : AND 01111111B
1067 : RET
1068 :
1069 :
1070 : SIOOUT: CP ' ' ;TEST FOR CONTROL CHARACTERS
1071 : JR NC,SIOXMT-$ ;JUMP IF PRINTABLE CHARACTER
1072 : CALL SIOXMT ;ELSE SEND CONTROL CHARACTER
1073 : LD A,(NULLS) ;AND THEN SEND NULLS AS FADDING
1074 : INC A ;GET NULL PAD COUNT AND FIX SO
1075 : JR PAD1-$ ;THAT COUNT=0 SENDS NO NULLS
1076 :
1077 : PAD:
1078 : PUSH AF
1079 : XDR A ;OUTPUT A NULL TO THE SIO
1080 : CALL POP AF
1081 : PAD1: DEC A
1082 : JR NZ,PAD-$ ;LOOP SENDING NULLS TO SIO
1083 : RET
1084 :
1085 :
1086 : SIOXMT: PUSH AF
1087 : SIOX1: IN A,(SIOCPB)

```

ADDRESS	INSTR	COMMENT	PC	PC+1	PC+2	PC+3	PC+4	PC+5	PC+6	PC+7	PC+8	PC+9	PC+10	PC+11	PC+12	PC+13	PC+14	PC+15	PC+16	PC+17	PC+18	PC+19	PC+20	PC+21	PC+22	PC+23	PC+24	PC+25	PC+26	PC+27	PC+28	PC+29	PC+30	PC+31	PC+32	PC+33	PC+34	PC+35	PC+36	PC+37	PC+38	PC+39	PC+40	PC+41	PC+42	PC+43	PC+44	PC+45	PC+46	PC+47	PC+48	PC+49	PC+50	PC+51	PC+52	PC+53	PC+54	PC+55	PC+56	PC+57	PC+58	PC+59	PC+60	PC+61	PC+62	PC+63	PC+64	PC+65	PC+66	PC+67	PC+68	PC+69	PC+70	PC+71	PC+72	PC+73	PC+74	PC+75	PC+76	PC+77	PC+78	PC+79	PC+80	PC+81	PC+82	PC+83	PC+84	PC+85	PC+86	PC+87	PC+88	PC+89	PC+90	PC+91	PC+92	PC+93	PC+94	PC+95	PC+96	PC+97	PC+98	PC+99	PC+100	PC+101	PC+102	PC+103	PC+104	PC+105	PC+106	PC+107	PC+108	PC+109	PC+110	PC+111	PC+112	PC+113	PC+114	PC+115	PC+116	PC+117	PC+118	PC+119	PC+120	PC+121	PC+122	PC+123	PC+124	PC+125	PC+126	PC+127	PC+128	PC+129	PC+130	PC+131	PC+132	PC+133	PC+134	PC+135	PC+136	PC+137	PC+138	PC+139	PC+140	PC+141	PC+142	PC+143	PC+144	PC+145	PC+146	PC+147	PC+148	PC+149	PC+150	PC+151	PC+152	PC+153	PC+154	PC+155	PC+156	PC+157	PC+158	PC+159	PC+160	PC+161	PC+162	PC+163	PC+164	PC+165	PC+166	PC+167	PC+168	PC+169	PC+170	PC+171	PC+172	PC+173	PC+174	PC+175	PC+176	PC+177	PC+178	PC+179	PC+180	PC+181	PC+182	PC+183	PC+184	PC+185	PC+186	PC+187	PC+188	PC+189	PC+190	PC+191	PC+192	PC+193	PC+194	PC+195	PC+196	PC+197	PC+198	PC+199	PC+200	PC+201	PC+202	PC+203	PC+204	PC+205	PC+206	PC+207	PC+208	PC+209	PC+210	PC+211	PC+212	PC+213	PC+214	PC+215	PC+216	PC+217	PC+218	PC+219	PC+220	PC+221	PC+222	PC+223	PC+224	PC+225	PC+226	PC+227	PC+228	PC+229	PC+230	PC+231	PC+232	PC+233	PC+234	PC+235	PC+236	PC+237	PC+238	PC+239	PC+240	PC+241	PC+242	PC+243	PC+244	PC+245	PC+246	PC+247	PC+248	PC+249	PC+250	PC+251	PC+252	PC+253	PC+254	PC+255	PC+256	PC+257	PC+258	PC+259	PC+260	PC+261	PC+262	PC+263	PC+264	PC+265	PC+266	PC+267	PC+268	PC+269	PC+270	PC+271	PC+272	PC+273	PC+274	PC+275	PC+276	PC+277	PC+278	PC+279	PC+280	PC+281	PC+282	PC+283	PC+284	PC+285	PC+286	PC+287	PC+288	PC+289	PC+290	PC+291	PC+292	PC+293	PC+294	PC+295	PC+296	PC+297	PC+298	PC+299	PC+300	PC+301	PC+302	PC+303	PC+304	PC+305	PC+306	PC+307	PC+308	PC+309	PC+310	PC+311	PC+312	PC+313	PC+314	PC+315	PC+316	PC+317	PC+318	PC+319	PC+320	PC+321	PC+322	PC+323	PC+324	PC+325	PC+326	PC+327	PC+328	PC+329	PC+330	PC+331	PC+332	PC+333	PC+334	PC+335	PC+336	PC+337	PC+338	PC+339	PC+340	PC+341	PC+342	PC+343	PC+344	PC+345	PC+346	PC+347	PC+348	PC+349	PC+350	PC+351	PC+352	PC+353	PC+354	PC+355	PC+356	PC+357	PC+358	PC+359	PC+360	PC+361	PC+362	PC+363	PC+364	PC+365	PC+366	PC+367	PC+368	PC+369	PC+370	PC+371	PC+372	PC+373	PC+374	PC+375	PC+376	PC+377	PC+378	PC+379	PC+380	PC+381	PC+382	PC+383	PC+384	PC+385	PC+386	PC+387	PC+388	PC+389	PC+390	PC+391	PC+392	PC+393	PC+394	PC+395	PC+396	PC+397	PC+398	PC+399	PC+400	PC+401	PC+402	PC+403	PC+404	PC+405	PC+406	PC+407	PC+408	PC+409	PC+410	PC+411	PC+412	PC+413	PC+414	PC+415	PC+416	PC+417
---------	-------	---------	----	------	------	------	------	------	------	------	------	------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------

(continued on top of page 10)

Micro Cornucopia, Number 2, September 1981

(continued next page)

PFM Monitor Listing

(continued)

```

F55C CBRF      1151 RES      7,A      ; SWITCH BACK LOWER 16K OF RAM
F55E D31C      1152 OUT      (BITDAT),A
F560 FB        1153 EI          ; INTERRUPTS ARE SAFE AGAIN
F561 C1        1154 POP        DE
F562 D1        1155 POP        DE
F563 E1        1156 POP        HL
F564 C9        1157 RET
F565          1158 ;
F566          1159 ;
F567          1160 ;
F568          1161 OUTCH: LD      DE,LEADIN
F569          1162 LD      A,(DE)
F570          1163 DR      A
F571          1164 JP      NZ,MULTI
F572          1165 LD      A,C
F573          1166 CP      ,
F574          1167 JR      C,CONTRL-#
F575          1168 DISPLA: LD    HL,(HL),C
F576          1169 INC      HL
F577          1170 LD      A,L
F578          1171 AND      01111111B
F579          1172 CP      B0
F580          1173 RET      C
F581          1174 CALL    RETURN
F582          1175 CALL    LFEEED
F583          1176 RET
F584          1177 ;
F585          1178 ;
F586          1179 ;
F587          1180 CONTRL: PUSH HL
F588          1181 LD      HL,CTLTAB
F589          1182 LD      BC,CTLSIZ/3;SEARCH FOR CONTROL CHARACTER
F590          1183 CALL    SEARCH
F591          1184 POP      HL
F592          1185 NZ      N2
F593          1186 PUSH    BC
F594          1187 RET
F595          1188 ;
F596          1189 CTLTAB: DEFB  ' ',-64
F597          1190 DEFB  ' ',-64
F598          1191 DEFB  ' ',-64
F599          1192 DEFB  'Z',-64
F600          1193 DEFB  'X',-64
F601          1194 DEFB  'Q',-64
F602          1195 DEFB  'M',-64
F603          1196 DEFB  'L',-64
F604          1197 DEFB  'K',-64
F605          1198 DEFB  'J',-64
F606          1199 DEFB  'I',-64
F607          1200 DEFB  'H',-64
F608          1201 DEFB  'G',-64
F609          1202 DEFB  ' ',-64
F610          1203 DEFW  BAKSPC
F611          1204 DEFW  TAB
F612          1205 DEFW  LFEEED
F613          1206 DEFW  LFEED
F614          1207 DEFW  UPSCR
F615          1208 DEFW  FORSPC
F616          1209 DEFW  BAKSPC
F617          1210 DEFW  TAB
F618          1211 DEFW  LFEEED
F619          1212 DEFW  LFEED
F620          1213 DEFW  UPSCR
F621          1214 DEFW  FORSPC
F622          1215 DEFW  BAKSPC
F623          1216 DEFW  TAB
F624          1217 DEFW  LFEEED
F625          1218 DEFW  LFEED
F626          1219 DEFW  UPSCR
F627          1220 DEFW  FORSPC
F628          1221 DEFW  BAKSPC
F629          1222 DEFW  TAB
F630          1223 DEFW  LFEEED
F631          1224 DEFW  LFEED
F632          1225 DEFW  UPSCR
F633          1226 DEFW  FORSPC
F634          1227 DEFW  BAKSPC
F635          1228 DEFW  TAB
F636          1229 DEFW  LFEEED
F637          1230 DEFW  LFEED
F638          1231 DEFW  UPSCR
F639          1232 DEFW  FORSPC
F640          1233 DEFW  BAKSPC
F641          1234 DEFW  TAB
F642          1235 DEFW  LFEEED
F643          1236 DEFW  LFEED
F644          1237 DEFW  UPSCR
F645          1238 DEFW  FORSPC
F646          1239 DEFW  BAKSPC
F647          1240 DEFW  TAB
F648          1241 DEFW  LFEEED
F649          1242 DEFW  LFEED
F650          1243 DEFW  UPSCR
F651          1244 DEFW  FORSPC
F652          1245 DEFW  BAKSPC
F653          1246 DEFW  TAB
F654          1247 DEFW  LFEEED
F655          1248 DEFW  LFEED
F656          1249 DEFW  UPSCR
F657          1250 DEFW  FORSPC
F658          1251 DEFW  BAKSPC
F659          1252 DEFW  TAB
F660          1253 DEFW  LFEEED
F661          1254 DEFW  LFEED
F662          1255 DEFW  UPSCR
F663          1256 DEFW  FORSPC
F664          1257 DEFW  BAKSPC
F665          1258 DEFW  TAB
F666          1259 DEFW  LFEEED
F667          1260 DEFW  LFEED
F668          1261 DEFW  UPSCR
F669          1262 DEFW  FORSPC
F670          1263 DEFW  BAKSPC
F671          1264 DEFW  TAB
F672          1265 DEFW  LFEEED
F673          1266 DEFW  LFEED
F674          1267 DEFW  UPSCR
F675          1268 DEFW  FORSPC
F676          1269 DEFW  BAKSPC
F677          1270 DEFW  TAB
F678          1271 DEFW  LFEEED
F679          1272 DEFW  LFEED
F680          1273 DEFW  UPSCR
F681          1274 DEFW  FORSPC
F682          1275 DEFW  BAKSPC
F683          1276 DEFW  TAB
F684          1277 DEFW  LFEEED
F685          1278 DEFW  LFEED
F686          1279 DEFW  UPSCR
F687          1280 DEFW  FORSPC
F688          1281 DEFW  BAKSPC
F689          1282 DEFW  TAB
F690          1283 DEFW  LFEEED
F691          1284 DEFW  LFEED
F692          1285 DEFW  UPSCR
F693          1286 DEFW  FORSPC
F694          1287 DEFW  BAKSPC
F695          1288 DEFW  TAB
F696          1289 DEFW  LFEEED
F697          1290 DEFW  LFEED
F698          1291 DEFW  UPSCR
F699          1292 DEFW  FORSPC
F700          1293 DEFW  BAKSPC
F701          1294 DEFW  TAB
F702          1295 DEFW  LFEEED
F703          1296 DEFW  LFEED
F704          1297 DEFW  UPSCR
F705          1298 DEFW  FORSPC
F706          1299 DEFW  BAKSPC
F707          1300 DEFW  TAB
F708          1301 DEFW  LFEEED
F709          1302 DEFW  LFEED
F710          1303 DEFW  UPSCR
F711          1304 DEFW  FORSPC
F712          1305 DEFW  BAKSPC
F713          1306 DEFW  TAB
F714          1307 DEFW  LFEEED
F715          1308 DEFW  LFEED
F716          1309 DEFW  UPSCR
F717          1310 DEFW  FORSPC
F718          1311 DEFW  BAKSPC
F719          1312 DEFW  TAB
F720          1313 DEFW  LFEEED
F721          1314 DEFW  LFEED
F722          1315 DEFW  UPSCR
F723          1316 DEFW  FORSPC
F724          1317 DEFW  BAKSPC
F725          1318 DEFW  TAB
F726          1319 DEFW  LFEEED
F727          1320 DEFW  LFEED
F728          1321 DEFW  UPSCR
F729          1322 DEFW  FORSPC
F730          1323 DEFW  BAKSPC
F731          1324 DEFW  TAB
F732          1325 DEFW  LFEEED
F733          1326 DEFW  LFEED
F734          1327 DEFW  UPSCR
F735          1328 DEFW  FORSPC
F736          1329 DEFW  BAKSPC
F737          1330 DEFW  TAB
F738          1331 DEFW  LFEEED
F739          1332 DEFW  LFEED
F740          1333 DEFW  UPSCR
F741          1334 DEFW  FORSPC
F742          1335 DEFW  BAKSPC
F743          1336 DEFW  TAB
F744          1337 DEFW  LFEEED
F745          1338 DEFW  LFEED
F746          1339 DEFW  UPSCR
F747          1340 DEFW  FORSPC
F748          1341 DEFW  BAKSPC
F749          1342 DEFW  TAB
F750          1343 DEFW  LFEEED
F751          1344 DEFW  LFEED
F752          1345 DEFW  UPSCR
F753          1346 DEFW  FORSPC
F754          1347 DEFW  BAKSPC
F755          1348 DEFW  TAB
F756          1349 DEFW  LFEEED
F757          1350 DEFW  LFEED
F758          1351 DEFW  UPSCR
F759          1352 DEFW  FORSPC
F760          1353 DEFW  BAKSPC
F761          1354 DEFW  TAB
F762          1355 DEFW  LFEEED
F763          1356 DEFW  LFEED
F764          1357 DEFW  UPSCR
F765          1358 DEFW  FORSPC
F766          1359 DEFW  BAKSPC
F767          1360 DEFW  TAB
F768          1361 DEFW  LFEEED
F769          1362 DEFW  LFEED
F770          1363 DEFW  UPSCR
F771          1364 DEFW  FORSPC
F772          1365 DEFW  BAKSPC
F773          1366 DEFW  TAB
F774          1367 DEFW  LFEEED
F775          1368 DEFW  LFEED
F776          1369 DEFW  UPSCR
F777          1370 DEFW  FORSPC
F778          1371 DEFW  BAKSPC
F779          1372 DEFW  TAB
F780          1373 DEFW  LFEEED
F781          1374 DEFW  LFEED
F782          1375 DEFW  UPSCR
F783          1376 DEFW  FORSPC
F784          1377 DEFW  BAKSPC
F785          1378 DEFW  TAB
F786          1379 DEFW  LFEEED
F787          1380 DEFW  LFEED
F788          1381 DEFW  UPSCR
F789          1382 DEFW  FORSPC
F790          1383 DEFW  BAKSPC
F791          1384 DEFW  TAB
F792          1385 DEFW  LFEEED
F793          1386 DEFW  LFEED
F794          1387 DEFW  UPSCR
F795          1388 DEFW  FORSPC
F796          1389 DEFW  BAKSPC
F797          1390 DEFW  TAB
F798          1391 DEFW  LFEEED
F799          1392 DEFW  LFEED
F800          1393 DEFW  UPSCR
F801          1394 DEFW  FORSPC
F802          1395 DEFW  BAKSPC
F803          1396 DEFW  TAB
F804          1397 DEFW  LFEEED
F805          1398 DEFW  LFEED
F806          1399 DEFW  UPSCR
F807          1400 DEFW  FORSPC
F808          1401 DEFW  BAKSPC
F809          1402 DEFW  TAB
F810          1403 DEFW  LFEEED
F811          1404 DEFW  LFEED
F812          1405 DEFW  UPSCR
F813          1406 DEFW  FORSPC
F814          1407 DEFW  BAKSPC
F815          1408 DEFW  TAB
F816          1409 DEFW  LFEEED
F817          1410 DEFW  LFEED
F818          1411 DEFW  UPSCR
F819          1412 DEFW  FORSPC
F820          1413 DEFW  BAKSPC
F821          1414 DEFW  TAB
F822          1415 DEFW  LFEEED
F823          1416 DEFW  LFEED
F824          1417 DEFW  UPSCR
F825          1418 DEFW  FORSPC
F826          1419 DEFW  BAKSPC
F827          1420 DEFW  TAB
F828          1421 DEFW  LFEEED
F829          1422 DEFW  LFEED
F830          1423 DEFW  UPSCR
F831          1424 DEFW  FORSPC
F832          1425 DEFW  BAKSPC
F833          1426 DEFW  TAB
F834          1427 DEFW  LFEEED
F835          1428 DEFW  LFEED
F836          1429 DEFW  UPSCR
F837          1430 DEFW  FORSPC
F838          1431 DEFW  BAKSPC
F839          1432 DEFW  TAB
F840          1433 DEFW  LFEEED
F841          1434 DEFW  LFEED
F842          1435 DEFW  UPSCR
F843          1436 DEFW  FORSPC
F844          1437 DEFW  BAKSPC
F845          1438 DEFW  TAB
F846          1439 DEFW  LFEEED
F847          1440 DEFW  LFEED
F848          1441 DEFW  UPSCR
F849          1442 DEFW  FORSPC
F850          1443 DEFW  BAKSPC
F851          1444 DEFW  TAB
F852          1445 DEFW  LFEEED
F853          1446 DEFW  LFEED
F854          1447 DEFW  UPSCR
F855          1448 DEFW  FORSPC
F856          1449 DEFW  BAKSPC
F857          1450 DEFW  TAB
F858          1451 DEFW  LFEEED
F859          1452 DEFW  LFEED
F860          1453 DEFW  UPSCR
F861          1454 DEFW  FORSPC
F862          1455 DEFW  BAKSPC
F863          1456 DEFW  TAB
F864          1457 DEFW  LFEEED
F865          1458 DEFW  LFEED
F866          1459 DEFW  UPSCR
F867          1460 DEFW  FORSPC
F868          1461 DEFW  BAKSPC
F869          1462 DEFW  TAB
F870          1463 DEFW  LFEEED
F871          1464 DEFW  LFEED
F872          1465 DEFW  UPSCR
F873          1466 DEFW  FORSPC
F874          1467 DEFW  BAKSPC
F875          1468 DEFW  TAB
F876          1469 DEFW  LFEEED
F877          1470 DEFW  LFEED
F878          1471 DEFW  UPSCR
F879          1472 DEFW  FORSPC
F880          1473 DEFW  BAKSPC
F881          1474 DEFW  TAB
F882          1475 DEFW  LFEEED
F883          1476 DEFW  LFEED
F884          1477 DEFW  UPSCR
F885          1478 DEFW  FORSPC
F886          1479 DEFW  BAKSPC
F887          1480 DEFW  TAB
F888          1481 DEFW  LFEEED
F889          1482 DEFW  LFEED
F890          1483 DEFW  UPSCR
F891          1484 DEFW  FORSPC
F892          1485 DEFW  BAKSPC
F893          1486 DEFW  TAB
F894          1487 DEFW  LFEEED
F895          1488 DEFW  LFEED
F896          1489 DEFW  UPSCR
F897          1490 DEFW  FORSPC
F898          1491 DEFW  BAKSPC
F899          1492 DEFW  TAB
F900          1493 DEFW  LFEEED
F901          1494 DEFW  LFEED
F902          1495 DEFW  UPSCR
F903          1496 DEFW  FORSPC
F904          1497 DEFW  BAKSPC
F905          1498 DEFW  TAB
F906          1499 DEFW  LFEEED
F907          1500 DEFW  LFEED
F908          1501 DEFW  UPSCR
F909          1502 DEFW  FORSPC
F910          1503 DEFW  BAKSPC
F911          1504 DEFW  TAB
F912          1505 DEFW  LFEEED
F913          1506 DEFW  LFEED
F914          1507 DEFW  UPSCR
F915          1508 DEFW  FORSPC
F916          1509 DEFW  BAKSPC
F917          1510 DEFW  TAB
F918          1511 DEFW  LFEEED
F919          1512 DEFW  LFEED
F920          1513 DEFW  UPSCR
F921          1514 DEFW  FORSPC
F922          1515 DEFW  BAKSPC
F923          1516 DEFW  TAB
F924          1517 DEFW  LFEEED
F925          1518 DEFW  LFEED
F926          1519 DEFW  UPSCR
F927          1520 DEFW  FORSPC
F928          1521 DEFW  BAKSPC
F929          1522 DEFW  TAB
F930          1523 DEFW  LFEEED
F931          1524 DEFW  LFEED
F932          1525 DEFW  UPSCR
F933          1526 DEFW  FORSPC
F934          1527 DEFW  BAKSPC
F935          1528 DEFW  TAB
F936          1529 DEFW  LFEEED
F937          1530 DEFW  LFEED
F938          1531 DEFW  UPSCR
F939          1532 DEFW  FORSPC
F940          1533 DEFW  BAKSPC
F941          1534 DEFW  TAB
F942          1535 DEFW  LFEEED
F943          1536 DEFW  LFEED
F944          1537 DEFW  UPSCR
F945          1538 DEFW  FORSPC
F946          1539 DEFW  BAKSPC
F947          1540 DEFW  TAB
F948          1541 DEFW  LFEEED
F949          1542 DEFW  LFEED
F950          1543 DEFW  UPSCR
F951          1544 DEFW  FORSPC
F952          1545 DEFW  BAKSPC
F953          1546 DEFW  TAB
F954          1547 DEFW  LFEEED
F955          1548 DEFW  LFEED
F956          1549 DEFW  UPSCR
F957          1550 DEFW  FORSPC
F958          1551 DEFW  BAKSPC
F959          1552 DEFW  TAB
F960          1553 DEFW  LFEEED
F961          1554 DEFW  LFEED
F962          1555 DEFW  UPSCR
F963          1556 DEFW  FORSPC
F964          1557 DEFW  BAKSPC
F965          1558 DEFW  TAB
F966          1559 DEFW  LFEEED
F967          1560 DEFW  LFEED
F968          1561 DEFW  UPSCR
F969          1562 DEFW  FORSPC
F970          1563 DEFW  BAKSPC
F971          1564 DEFW  TAB
F972          1565 DEFW  LFEEED
F973          1566 DEFW  LFEED
F974          1567 DEFW  UPSCR
F975          1568 DEFW  FORSPC
F976          1569 DEFW  BAKSPC
F977          1570 DEFW  TAB
F978          1571 DEFW  LFEEED
F979          1572 DEFW  LFEED
F980          1573 DEFW  UPSCR
F981          1574 DEFW  FORSPC
F982          1575 DEFW  BAKSPC
F983          1576 DEFW  TAB
F984          1577 DEFW  LFEEED
F985          1578 DEFW  LFEED
F986          1579 DEFW  UPSCR
F987          1580 DEFW  FORSPC
F988          1581 DEFW  BAKSPC
F989          1582 DEFW  TAB
F990          1583 DEFW  LFEEED
F991          1584 DEFW  LFEED
F992          1585 DEFW  UPSCR
F993          1586 DEFW  FORSPC
F994          1587 DEFW  BAKSPC
F995          1588 DEFW  TAB
F996          1589 DEFW  LFEEED
F997          1590 DEFW  LFEED
F998          1591 DEFW  UPSCR
F999          1592 DEFW  FORSPC

```

F5A8 E7F5	1209	DEFW	RETURN	CTL-M IS <CR>	F645 17	1338	RLA	:EXTRACT ROW# COMPONENT OF HL
F5AA 11F6	1210	DEFW	CLREDS	:CTL-Q CLEAR TO END-OF-SCREEN	F646 E61F	1339	AND	
F5AC 03F6	1211	DEFW	CLREOL	:CTL-X IS CLEAR TO END-OF-LINE	F648 4F	1340	LD	:COPY ROW# TO C FOR SCROLL TEST
F5AE E6F5	1212	DEFW	CLRSN	:CTL-Z IS CLEAR SCREEN	F649 CD37F6	1341	CALL	:MOVE CURSOR TO NEXT ROW DOWN
F5B0 B6F5	1213	DEFW	ESCAPE	:CTL-1 IS ESCAPE	F64C 3A77FF	1342	LD	:TEST IF CURSOR ON BOTTOM ROW
F5B2 60F6	1214	DEFW	HOMEUP	:CTL-^ IS HOME UP	F64F B9	1343	CP	:OF SCREEN BEFORE MOVING DOWN
F5B4 BAF5	1215	DEFW	STUFF	:CTL- _ IS DISPLAY CONTROL CHARS	F650 C0	1344	RET	:EXIT IF NOT AT BOTTOM
>0027	1216	CTL512 EQU	\$-CTLTAB		F651 E5	1345		
	1217	CTL512 EQU			F652 CD60F6	1346	PUSH	:ELSE PREP TO SCROLL SCREEN UP
	1218				F653 29	1347	CALL	:FILL NEW BOTTOM LINE WTH SPACES
F5B6 3E01	1219	ESCAPE: LD	A,1		F656 7C	1349	ADD	
F5B8 12	1220	LD	(DE),A	:SET LEAD-IN SEQUENCE STATE	F657 E61F	1350	AND	:SET ROW# PART OF HL INTO A
F5B9 C9	1221	LD		:FOR XY CURSOR POSITIONING MODE	F659 3277FF	1351	LD	:STORE NEW BASE LINE#
	1222	RET			F65C D314	1352	OUT	:SCROLL UP NEW BLANK BOTTM LINE
	1223				F65E E1	1353	POP	
F5BA 3E04	1224	STUFF: LD	A,4		F65F C9	1354	RET	
F5BC 12	1225	LD	(DE),A	:SET LEAD-IN SEQUENCE STATE		1355		
F5BD C9	1226	LD		:FOR CONTROL CHAR OUTPUT MODE	F660 7D	1356	LD	
	1227	RET			F661 E480	1357	AND	
	1228				F663 6F	1358	LD	:POINT HL TO 1ST COLUMN OF ROW
F5BE 7D	1229	BAKSPC	A,L	:CHECK FOR LEFT MARGIN	F664 0650	1359	LD	
F5BF E67F	1230	AND	01111111B		F666 3620	1361	LD	:STORE ASCII SPACES AT ADDR
F5C1 C8	1231	RET	Z	:ABORT IF IN LEFTMOST COLUMN				IN HL
F5C2 2B	1232	DEC	HL	:BACK UP CURSOR POINTER	F668 23	1362	INC	:AND INCREMENT HL
F5C3 C9	1233	RET			F669 10FB	1363	DJNZ	:REPEAT NUMBER OF TIMES IN B
	1234				F66B C9	1364	RET	
	1235					1365		
F5C4 7D	1236		A,L	:CHECK FOR RIGHTMOST COLUMN	F66C 0E20	1366	LD	
F5C5 E67F	1237	FORSPC: LD	01111111B		F66E 1B17	1367	LD	:FAKE-OUT CURSOR ADDR ROUTINE
F5C7 FE4F	1238	AND	79	:DO NOTHING IF ALREADY THERE		1368	JR	:TO DO HOMEUP ALMOST FOR FREE
F5C9 D0	1239	CP	NC	:ELSE ADVANCE CURSOR POINTER		1369		
F5CA 23	1240	INC	HL			1370		
F5CB C9	1241	RET			F670 EB	1371	EX	
	1242				F671 3600	1372	LD	:UNCONDITIONALLY RESET LEAD-IN
	1243				F673 EB	1373	EX	:STATE TO ZERO BEFORE GOING ON
	1244				F674 FE01	1374	CP	
F5CC 110600	1245	TAB:	DE,8	:TABS ARE EVERY 8 COLUMNS	F676 200B	1375	JR	
F5CF 7D	1246	LD	A,L	:SET COLUMN COMPONENT OF	F678 79	1376	SETXY: LD	:GET SECOND CHAR OF SEQUENCE
F5D0 E678	1247	AND	01111000B	:PREVIOUS TAB POSITION	F679 FED	1377	CP	
F5D2 B3	1248	ADD	A,E	:EXIT IF NEXT TAB COLUMN WOULD	F67B C0	1378	LD	:ABORT SEQUENCE IF NOT '='
F5D3 FE50	1249	CP	80	:BE PAST THE RIGHT MARGIN	F67C 3E02	1379	LD	:MAKE LEADIN=2 NEXT TIME
F5D5 D0	1250	CP	NC		F67E 12	1380	LD	
F5D6 7D	1251	LD	A,L	:ELSE INCREMENT THE CURSOR	F67F C9	1381	RET	
F5D7 E678	1252	AND	11111000B	:POINTER FOR REAL		1382		
F5D9 6F	1253	LD	L,A		F680 FE02	1383	CP	
F5DA 19	1254	ADD	HL,DE		F682 2019	1384	JR	
F5DB C9	1255	RET			F684 3E03	1385	LD	
	1256				F686 12	1386	LD	
F5DC DB1C	1257	BELL:	A,(BITDAT)	:TOGGLE BIT 5 OF SYSTEM PIO TO	F687 3A77FF	1387	LD	:MAKE LEADIN=3 NEXT TIME
F5DE DBEF	1258	IN	S,A	:TRIGGER BELL HARDWARE TO SOUND	F68A 81	1388	LD	:ARRIVE HERE ON THIRD CHAR
F5E0 D31C	1259	SET	(BITDAT),A		F68B D61F	1389	ADD	:OF ESC, '-', ROW,COL SEQUENCE
F5E2 CB4F	1260	OUT	S,A		F68F 30FC	1391	SUB	
F5E4 D31C	1261	RES	(BITDAT),A		F691 C61B	1392	JR	
F5E6 C9	1262	OUT			F693 F660	1393	OR	
	1263	RET			F695 67	1394	LD	
	1264				F696 2E00	1395	LD	
F5E7 7D	1265	RETURN: LD	A,L	:MOVE CURSOR POINTER BACK	F698 CB3C	1396	SRL	
F5E8 E680	1266	AND	10000000B	:TO START OF LINE	F69A DB1D	1397	RR	
F5EA 6F	1267	LD	L,A		F69C C9	1398	RET	
F5EB C9	1268	RET				1399		
	1269				F69D FE03	1400	CP	
	1270				F69F 200C	1401	JR	
F5EC 210030	1271		HL,CRTHMEM					
	1272	CLRSN: LD						

(continued next page)

(continued on top of page 12)

PFM Monitor Listing

(continued)

```

F6A1 79 1402 SETCOL: LD A,C ;ARRIVE HERE ON FOURTH CHAR
F6A2 D620 1403 SUB ; OF ESC, '=', ROW,COL SEQUENCE
F6A4 D650 1404 SETC2: SUB 80
F6A6 30FC 1405 JR NC,SETC2-# ;MAKE SURE COL# BETWEEN 0 & 79
F6A8 C650 1406 ADD A,B0
F6AA B5 1407 OR L
F6AB 6F 1408 LD L,A
F6AC C9 1409 RET
F6AD CD72F5 1410
F6B0 C9 1411 MATST: CALL DISPLA ;DISPLAY THE CONTROL CHAR
1412 RET ;PASSED IN C
1413 ;
1414 ;
1415 ;
1416 ;
1417 ;
1418 ;***** INCLUDE DISKIO.ASM *****
1419 ;*****
1420 ; DISK INPUT/OUTPUT DRIVER SUBROUTINE PACKAGE
1421 ; FOR WESTERN DIGITAL 1771 DISK CONTROLLER
1422 ;
1423 ;
1424 ; bullet-proof error recovery added 12-APR-80
1425 ;*****
1426 ;
1427 ;
1428 ;EQUATES FOR DISK CONTROLLER PORTS AND COMMAND CODES
1429 ;
1430 STREG EQU WD1771+0 ;STATUS REGISTER
1431 CMDREG EQU WD1771+0 ;COMMAND REGISTER
1432 TRKREG EQU WD1771+1 ;TRACK REGISTER
1433 SECREG EQU WD1771+2 ;SECTOR REGISTER
1434 DATREG EQU WD1771+3 ;DATA REGISTER
1435 ;
1436 RDCMD EQU 10001000B ;READ COMMAND
1437 WRTCMD EQU 10101000B ;WRITE COMMAND
1438 SKCMD EQU 00011100B ;SEEK COMMAND
1439 FINCMD EQU 11010000B ;FORCE INTR COMMAND
1440 RSTCMD EQU 00001100B ;RESTORE COMMAND
1441 HLOAD EQU 000000100B ;RD/WRT HEAD LOAD ENABLE
1442 ;
1443 RET EQU 0C9H ;SUBROUTINE RETURN INSTR OPCODE
1444 NMIVEC EQU 0066H ;THE NON-MASKABLE INTERRUPT IS
1445 ;USED FOR DATA SYNC BETWEEN
1446 ;THE Z-80 AND 1771
1447 ;
1448 ;
1449 ;
1450 SELECT: LD A,C ;GET UNIT# PASSED IN C AND
1451 CP 4 ; CHECK FOR MAXIMUM VALID#
1452 RET NC ;ERROR IF NUMBER > 3
1453 CALL TURNON ;MAKE SURE DISKS ARE TURNED ON
1454 IN A,(BITDAT)
1455 LD B,A ;SAVE CURRENT DRIVE SELECT DATA
1456 AND 11111000B ;MERGE IN NEW DRIVE UNIT# IN C
1457 OR C ;IN PLACE OF THE CURRENT ONE
1458 OUT (BITDAT),A ;TO SELECT THE NEW DISK DRIVE
1459 CALL FORCE ;TEST NEW DRIVE'S READY STATUS
F6B1 79
F6B2 FE04
F6B4 D0
F6B5 CDB8F7
F6B8 DB1C
F6BA 47
F6BB E6F8
F6BD B1
F6BE D31C
F6BF CDAEF7

```


PFM Monitor Listing (continued)

```

1651 ;*****
1652 ;*
1653 ;*   STORAGE ALLOCATION FOR 256 BYTE SCRATCH RAM   *
1654 ;*
1655 ;*****
1656 ;
1657 ;
1658 ;
1659 VECTAB EQU $
1660 STOVEC: DEFS 16
1661 CTCVEC: DEFS 8
1662 SYSVEC: DEFS 4
1663 GENVEC: DEFS 4
1664 ;
1665 ;
1666 ;KEYBOARD DATA INPUT FIFO VARIABLES
1667
1668 FIFO: DEFS 16
1669 FIFONT: DEFS 1
1670 FIFIN: DEFS 1
1671 FIFOUT: DEFS 1
1672 LOCK: DEFS 2
1673 ;
1674 ;
1675 ;STACK POINTER SAVE AND LOCAL STACK FOR INTERRUPT ROUTINES
1676
1677 SPSAVE: DEFS 2
1678 TMPSTK: DEFS 32
1679 ;
1680 ;
1681 ;'SOFTWARE' VECTORS FOR INTERRUPT SERVICE ROUTINES
1682
1683 TIKVEC: DEFS 2
1684 PINVEC: DEFS 2
1685 SINVEC: DEFS 2
1686 ;
1687 ;
1688 ;CLOCK-TIMER INTERRUPT VARIABLES
1689
1690 TIKCNT: DEFS 2
1691 DAY: DEFS 1
1692 MONTH: DEFS 1
1693 YEAR: DEFS 1
1694 HRS: DEFS 1
1695 MINS: DEFS 1
1696 SECS: DEFS 1
1697 ;
1698 ;
1699 ;DISK I/O DRIVER VARIABLES
1700
1701 UNIT: DEFS 1
1702 TRKTAB: DEFS 4
1703 SPEED: DEFS 1
1704 RELEN: DEFS 1
1705 MOTOR: DEFS 1
1706 TRACK: DEFS 1
1707 SECTOR: DEFS 1
1708 ;
1709 ;
1710 ;CURRENTLY SELECTED DISK#
1711 ;
1712 ;4 DRIVE HEAD POSITION TABLE
1713 ;
1714 ;SEEK SPEED FOR 1771 COMMANDS
1715 ;
1716 ;SECTOR RECORD LENGTH VARIABLE
1717 ;
1718 ;DRIVE MOTOR TURN-OFF TIMER
1719 ;
1720 ;

```

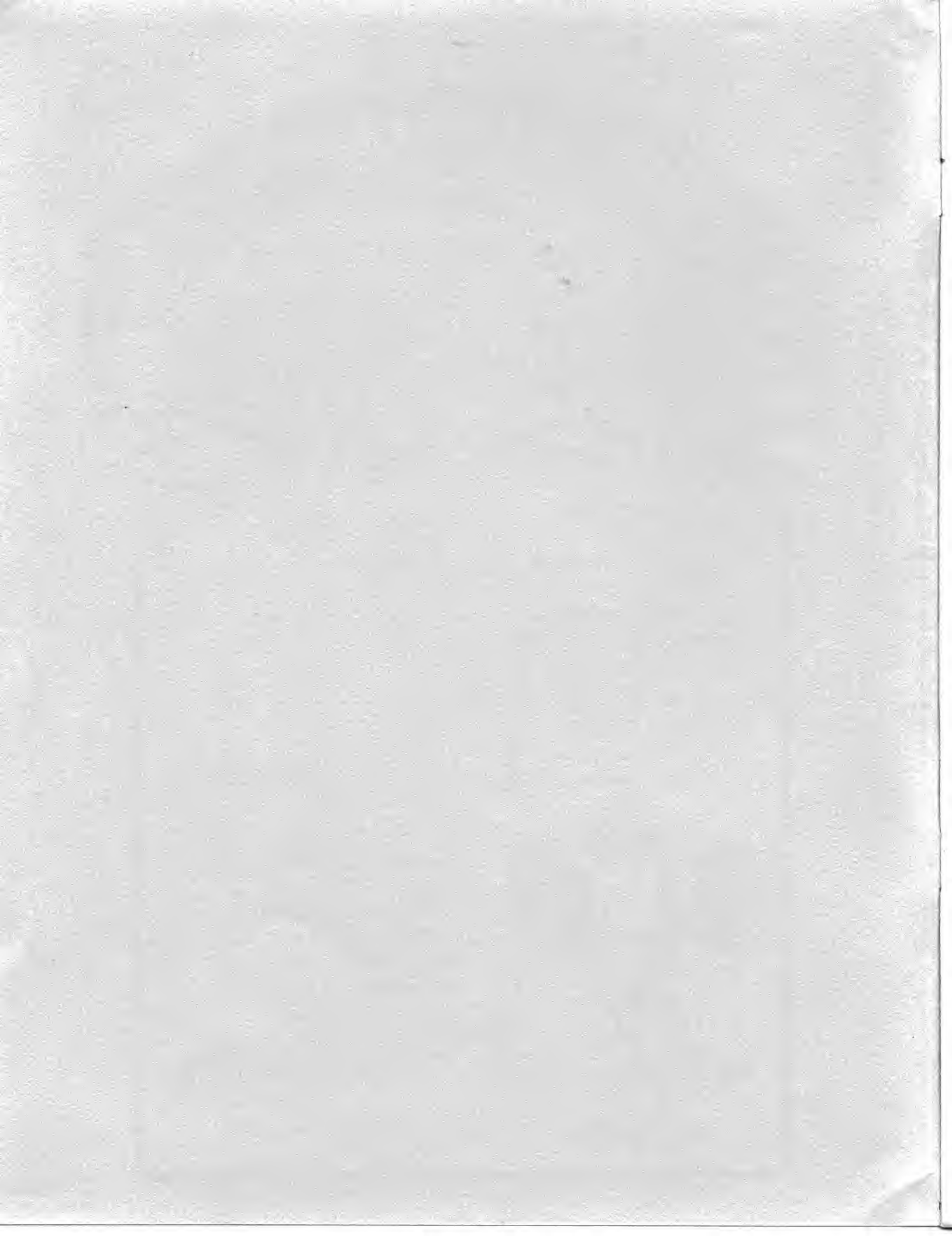
```

>FF6F CMTYP: DEFS 1
>FF70 RETRY: DEFS 1
>FF71 IOPTR: DEFS 2
1711 ;
1712 ;
1713 ;
1714 ;CRT OUTPUT DRIVER VARIABLES
1715
1716 CURSOR: DEFS 2
1717 CHRSAV: DEFS 1
1718 CSRCHR: DEFS 1
1719 BASE: DEFS 1
1720 LEADIN: DEFS 1
1721 ;
1722 ;
1723 ;NULL PAD COUNT FOR SERIAL OUTPUT DELAY
1724
1725 NULLS: DEFS 1
1726 ;
1727 ;
1728 ;LISTHEAD POINTER FOR DYNAMIC MEMORY ALLOCATION SCHEME
1729
1730 FREPTR: DEFS 2
1731 ;
1732 ;
1733 ;CONSOLE MONITOR PROGRAM VARIABLES
1734
1735 PARAM1: DEFS 2
1736 PARAM2: DEFS 2
1737 PARAM3: DEFS 2
1738 PARAM4: DEFS 2
1739 ESCFLG: DEFS 1
1740 COFLAG: DEFS 1
1741 LAST: DEFS 2
1742 LINBUF: DEFS 64
1743 ;
1744 ;
1745 ;
1746 ;
1747 ;
1748 ;
1749 ;
1750 ;
1751 ;
1752 ;
1753 ;
1754 ;
1755 ;
1756 ;
1757 ;
1758 ;
1759 ;
1760 ;
1761 ;
1762 ;
1763 ;
1764 ;
1765 ;
1766 ;
1767 ;
1768 ;
1769 ;
1770 ;
1771 ;
1772 ;
1773 ;
1774 ;
1775 ;
1776 ;
1777 ;
1778 ;
1779 ;
1780 ;
1781 ;
1782 ;
1783 ;
1784 ;
1785 ;
1786 ;
1787 ;
1788 ;
1789 ;
1790 ;
1791 ;
1792 ;
1793 ;
1794 ;
1795 ;
1796 ;
1797 ;
1798 ;
1799 ;
1800 ;
1801 ;
1802 ;
1803 ;
1804 ;
1805 ;
1806 ;
1807 ;
1808 ;
1809 ;
1810 ;
1811 ;
1812 ;
1813 ;
1814 ;
1815 ;
1816 ;
1817 ;
1818 ;
1819 ;
1820 ;
1821 ;
1822 ;
1823 ;
1824 ;
1825 ;
1826 ;
1827 ;
1828 ;
1829 ;
1830 ;
1831 ;
1832 ;
1833 ;
1834 ;
1835 ;
1836 ;
1837 ;
1838 ;
1839 ;
1840 ;
1841 ;
1842 ;
1843 ;
1844 ;
1845 ;
1846 ;
1847 ;
1848 ;
1849 ;
1850 ;
1851 ;
1852 ;
1853 ;
1854 ;
1855 ;
1856 ;
1857 ;
1858 ;
1859 ;
1860 ;
1861 ;
1862 ;
1863 ;
1864 ;
1865 ;
1866 ;
1867 ;
1868 ;
1869 ;
1870 ;
1871 ;
1872 ;
1873 ;
1874 ;
1875 ;
1876 ;
1877 ;
1878 ;
1879 ;
1880 ;
1881 ;
1882 ;
1883 ;
1884 ;
1885 ;
1886 ;
1887 ;
1888 ;
1889 ;
1890 ;
1891 ;
1892 ;
1893 ;
1894 ;
1895 ;
1896 ;
1897 ;
1898 ;
1899 ;
1900 ;
1901 ;
1902 ;
1903 ;
1904 ;
1905 ;
1906 ;
1907 ;
1908 ;
1909 ;
1910 ;
1911 ;
1912 ;
1913 ;
1914 ;
1915 ;
1916 ;
1917 ;
1918 ;
1919 ;
1920 ;
1921 ;
1922 ;
1923 ;
1924 ;
1925 ;
1926 ;
1927 ;
1928 ;
1929 ;
1930 ;
1931 ;
1932 ;
1933 ;
1934 ;
1935 ;
1936 ;
1937 ;
1938 ;
1939 ;
1940 ;
1941 ;
1942 ;
1943 ;
1944 ;
1945 ;
1946 ;
1947 ;
1948 ;
1949 ;
1950 ;
1951 ;
1952 ;
1953 ;
1954 ;
1955 ;
1956 ;
1957 ;
1958 ;
1959 ;
1960 ;
1961 ;
1962 ;
1963 ;
1964 ;
1965 ;
1966 ;
1967 ;
1968 ;
1969 ;
1970 ;
1971 ;
1972 ;
1973 ;
1974 ;
1975 ;
1976 ;
1977 ;
1978 ;
1979 ;
1980 ;
1981 ;
1982 ;
1983 ;
1984 ;
1985 ;
1986 ;
1987 ;
1988 ;
1989 ;
1990 ;
1991 ;
1992 ;
1993 ;
1994 ;
1995 ;
1996 ;
1997 ;
1998 ;
1999 ;
2000 ;

```

ERRORS=0000

end



MICRO CORNUCOPIA

Journal of the Big Board Users Group

P.O. BOX 223
BEND, OREGON 97709